**[First Person Shooter] TDD**

[version number v\_xx.xx]

Author: [Ethan McGimpsey]

**Introduction**

Rationale

What I am trying to accomplish is a working FPS game that mainly focuses around on a gun from the players perspective, such as functional animations, shooting mechanics and ragdolls that would fit as a base to any first person shooter games. We also accomplished adding multiplayer elements to the game where you can connect with upto 4 players and battle each other.

Background

First-person shooter is a video game genre centered on gun and other weapon-based combat in a first-person perspective; that is, the player experiences the action through the eyes of the protagonist. The genre shares common traits with other shooter games, which in turn makes it fall under the heading action game.

Terminology

**First Person Shooter** - Involves shooting enemies and other targets and in which a player views the action as though through the eyes of the character they are controlling.

**Shooter** - Has weapons in game that players can use to kill enemies or kill other players.

Non-Goals

Not applicable

Proposed Design

Players will have the choice to play one of two game modes: Capture the Flag mode and One vs One mode. The gamemode that I am currently using is the One vs One mode (CTF) that has a grey-boxed layout of the map, it was created using Unity's Pro Builder.

**System Architecture**



Data types

* **Floats**: The floats were used in a couple of our scripts and they were for how much damage, range, firerate, impactforce and reloadtime the weapon has. How much health does the player have, how much movement speed and how much gravity would the player have.
* **Bool**: The bool was used in one script and it was only used for to check if the player is dead. Check if the player is touching the ground so that you are able to jump again, check if they're firing, reloading or zooming in. And check if the enemy is dead when the health is < 0.
* **GameObject**: The GameObjects were used in a couple of scripts and they were for the scoreText, Health UI text, healthCanvas for when the enemie loses health.
* **Sprite**[]: The Sprites were used in one script and it was only used for to change the sprite everytime when the player goes through a column.
* **Int**: The int was used in a couple of our scripts and they were for the amount of bullets the weapon has, setting the difficulity for the enemies and setting the curwaypoint to start on.
* **Vector3**: The Vector3 type was used for tracking the players position and setting the drop location for the flag to spawn.
* **Rigidbody2D**: The Rigidbody2D was used in one script and it was only used for to give gravity to the player.

Data Model

The Data is being stored in the GameMode script. How it works is everytime when a player captures a flag and brings it back to base, they will earn a scorepoint for succesfully capturing it and returning it back to base.

Interface/API Definitions

Since Unity's Networkign API depreciated, Mirror has been the most useful tool to use for networking. With Mirror, it optimizes the ease of use and probability of success, mirror makes networking easy, concise and maintainable.

Impact

The project was intended to arrive at most extreme execution capacity while delivering a great 90s feel towards the FPS kind roused by the dad of FPS' DOOM and Wolfenstein. With no tangled code and each class isolated inside their information types, this undertaking will run on practically any PC arriving at greatest casings every second without taking a lot of capacity when introduced.

Risks

There were no risks or unknowns that occurred in or during this project.

Alternatives

I had a solution that could be useful to the game, here are my solutions that I came up with but were rejected:

* Multiple guns and weapons
* After a team wins a match, the match will reset.

**System Architecture**

Testing

(See upload Test Plan Document for documented testing using Unity's Unit Testing)